KIP-358: Migrate Streams API to Duration instead of long ms times

Status

Current state: *Accepted*

Discussion thread: here

JIRA: KAFKA-7277 - Getting issue details... STATUS , KAFKA-7477 - Getting issue details... STATUS

Motivation

1. KAFKA-7277:

   Right now Streams API universally represents time as ms-since-unix-epoch.
   There's nothing wrong, per se, with this, but Duration is more ergonomic for an API.
   What we don't want is to present a heterogeneous API, so we need to make sure the whole Streams API is in terms of Duration.

2. KAFKA-7477:
   Improve of semantics of `KafkaStreams#close(Duration)`

Public Interfaces

We need to add following method to public API.

Some of the old methods become deprecated. Other can be replaced, because they are not released, yet.

Method that proposed for replacement added in KAFKA-7222 - Getting issue details... STATUS (KIP-328, commit b3771ba):

Public API changes

```
class Window {
        //New methods.
        Instant startTime();
        Instant endTime();
}

JoinWindows {
    //Existing methods. Will be deprecated.
    static JoinWindows of(final long timeDifferenceMs) throws IllegalArgumentException;
    JoinWindows before(final long timeDifferenceMs) throws IllegalArgumentException;
    JoinWindows after(final long timeDifferenceMs) throws IllegalArgumentException;

        //Existing method. Will be removed.
        JoinWindows grace(final long millisAfterWindowEnd);

    //New methods.
    static JoinWindows of(final Duration timeDifference) throws IllegalArgumentException;
    JoinWindows before(final Duration timeDifference) throws IllegalArgumentException;
    JoinWindows after(final Duration timeDifference) throws IllegalArgumentException;
        JoinWindows grace(final Duration afterWindowEnd) throws IllegalArgumentException;
}

Materialized {
        //Existing method. Will be removed.
        Materialized<K, V, S> withRetention(final long retentionMs);
```

```
        //New method.
        Materialized<K, V, S> withRetention(final Duration retention) throws IllegalArgumentException;
}

SessionWindows {
        //Existing methods. Will be deprecated.
    static SessionWindows with(final long inactivityGapMs);

        //Existing methods. Will be removed.
        SessionWindows grace(final long millisAfterWindowEnd);

        //New methods.
    static SessionWindows with(final Duration inactivityGap) throws IllegalArgumentException;
        SessionWindows grace(final Duration afterWindowEnd) throws IllegalArgumentException;
}

TimeWindows {
        //Existing methods. Will be deprecated.
        static TimeWindows of(final long sizeMs) throws IllegalArgumentException;
        TimeWindows advanceBy(final long advanceMs);

        //Existing method. Will be removed.
        TimeWindows grace(final long millisAfterWindowEnd);

        //New methods.
        static TimeWindows of(final Duration size) throws IllegalArgumentException;
        TimeWindows advanceBy(final Duration advance);
        TimeWindows grace(final Duration afterWindowEnd);
}

UnlimitedWindows {
        //Existing methods. Will be deprecated.
    UnlimitedWindows startOn(final long startMs) throws IllegalArgumentException;

        //New methods.
    UnlimitedWindows startOn(final Instant start) throws IllegalArgumentException;
}

ProcessorContext {
    //Existing method. Will be deprecated.
    Cancellable schedule(final long intervalMs,
                         final PunctuationType type,
                         final Punctuator callback);

        //New method.
    Cancellable schedule(final Duration interval,
                         final PunctuationType type,
                         final Punctuator callback) throws IllegalArgumentException;
}

ReadOnlyWindowStore<K, V> {
    //Deprecated methods.
    WindowStoreIterator<V> fetch(K key, long timeFrom, long timeTo);
    KeyValueIterator<Windowed<K>, V> fetch(K from, K to, long timeFrom, long timeTo);
    KeyValueIterator<Windowed<K>, V> fetchAll(long timeFrom, long timeTo);

        //New methods.
        //This changed after initial KIP voting based on [PR discussion](https://github.com/apache/kafka/pull
/5682#discussion_r222494244)
    WindowStoreIterator<V> fetch(K key, Instant from, Instant to) throws IllegalArgumentException;
    KeyValueIterator<Windowed<K>, V> fetch(K from, K to, Instant from, Instant to) throws
IllegalArgumentException;
    KeyValueIterator<Windowed<K>, V> fetchAll(Instant from, Instant to) throws IllegalArgumentException;
}

WindowStore {
        //New methods with default implementation that checks arguments and pass it to existing fetch methods.
    WindowStoreIterator<V> fetch(K key, long timeFrom, long timeTo) throws IllegalArgumentException;
    KeyValueIterator<Windowed<K>, V> fetch(K from, K to, long timeFrom, long timeTo) throws
IllegalArgumentException;
```

```
    KeyValueIterator<Windowed<K>, V> fetchAll(long timeFrom, long timeTo) throws IllegalArgumentException;
}

Stores {
        //Existing methods. Will be deprecated.
        static WindowBytesStoreSupplier persistentWindowStore(final String name,
                                                              final long retentionPeriodMs,
                                                              final long windowSizeMs,
                                                            final boolean retainDuplicates);


        //This method added after KIP voting. Based on John Roesler comment(https://github.com/apache/kafka/pull
/5682#discussion_r221472187)
        static SessionBytesStoreSupplier persistentSessionStore(final String name,
                                                               final long retentionPeriod);

        static WindowBytesStoreSupplier persistentWindowStore(final String name,
                                                            final long retentionPeriodMs,
                                                            final long windowSizeMs,
                                                            final boolean retainDuplicates,
                                                            final long segmentIntervalMs);

        //New methods.
        static WindowBytesStoreSupplier persistentWindowStore(final String name,
                                                              final Duration retentionPeriod,
                                                              final Duration windowSize,
                                                            final boolean retainDuplicates) throws
IllegalArgumentException;

        static SessionBytesStoreSupplier persistentSessionStore(final String name,
                                                                final Duration retentionPeriod) throws
IllegalArgumentException;
}

KafkaStreams {
        //Existing method. Will be deprecated.
        public synchronized boolean close(final long timeout, final TimeUnit timeUnit);

        //New method
        public synchronized boolean close(final Duration timeout) throws IllegalArgumentException;
}
```

Proposed Changes

New methods in public API are proposed. See "Public Interfaces" section.

For the methods that used both: internally and as a part of public API the proposal is:

1. In this scope keep existing methods as is.
   Try to reduce the visibility of methods in next tickets.
2. Introduce finer methods with `Instant` and `Duration`

Changes in **`KafkaStream#close`** semantics:

1. reject negative numbers
2. make 0 just signal and return immediately (after checking the state once)

Default implementation of fetch methods in WindowStore.

Compatibility, Deprecation, and Migration Plan

No compatibility issues foreseen.

Rejected Alternatives

An alternative solution with long parameters is implemented right now.